LabVIEW™

PDA Module User Manual

April 2003 Edition
Part Number 370624A-01

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 683 0100

Worldwide Offices

Australia 02 612 9672 8846, Austria 43 0 662 45 79 90 0, Belgium 32 0 2 757 00 20, Brazil 55 11 3262 3599,
Canada (Calgary) 403 274 9391, Canada (Montreal) 514 288 5722, Canada (Ottawa) 613 233 5949,
Canada (Québec) 514 694 8521, Canada (Toronto) 905 785 0085, Canada (Vancouver) 514 685 7530,
China 86 21 6555 7838, Czech Republic 420 2 2423 5774, Denmark 45 45 76 26 00,
Finland 385 0 9 725 725 11, France 33 0 1 48 14 24 24, Germany 49 0 89 741 31 30, Greece 30 2 10 42 96 427,
India 91 80 51190000, Israel 972 0 3 6393737, Italy 39 02 413091, Japan 81 3 5472 2970,
Korea 82 02 3451 3400, Malaysia 603 9131 0918, Mexico 001 800 010 0793, Netherlands 31 0 348 433 466,
New Zealand 64 09 914 0488, Norway 47 0 32 27 73 00, Poland 48 0 22 3390 150, Portugal 351 210 311 210,
Russia 7 095 238 7139, Singapore 65 6226 5886, Slovenia 386 3 425 4200, South Africa 27 0 11 805 8197,
Spain 34 91 640 0085, Sweden 46 0 8 587 895 00, Switzerland 41 56 200 51 51, Taiwan 886 2 2528 7227,
Thailand 662 992 7519, United Kingdom 44 0 1635 523545

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on the documentation, send email to techpubs@ni.com.

Important Information

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Trademarks

LabVIEW™, National Instruments™, NI™, and ni.com™ are trademarks of National Instruments Corporation.

Product and company names mentioned herein are trademarks or trade names of their respective companies.

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your CD, or `ni.com/patents`.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Contents

Chapter 4
Building Wireless PDA Applications

Chapter 5
Programming Techniques

Appendix A
Technical Support and Professional Services

About This Manual

This manual contains introductory information about the LabVIEW PDA Module and describes how to use the PDA Module.

Refer to the *LabVIEW PDA Module Release Notes* for installation instructions, getting started information, and a tutorial that guides you through the basics of building, running, and debugging PDA VIs.

This manual assumes you are familiar with Microsoft Windows and basic LabVIEW concepts. This manual also assumes you possess basic knowledge of Palm OS and/or Pocket PC 2002.

Conventions

The following conventions appear in this manual:

»	The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.
	This icon denotes a note, which alerts you to important information.
	This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash.
bold	Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.
italic	Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept. This font also denotes text that is a placeholder for a word or value that you must supply.
`monospace`	Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames and extensions, and code excerpts.
`monospace italic`	Italic text in this font denotes text that is a placeholder for a word or value that you must supply.

Platform	Text in this font denotes a specific platform, such as Palm OS or Pocket PC 2002, and indicates that the text following it applies only to that platform.
tap	A tap on a PDA device is the equivalent to a mouse click on a desktop computer.

Related Documentation

The following documents contain information that you might find helpful as you read this manual:

- *LabVIEW PDA Module Release Notes*
- *LabVIEW User Manual*, available in PDF format by selecting **Start»Programs»National Instruments»LabVIEW 7.0»Search the LabVIEW Bookshelf** and clicking the LabVIEW User Manual link
- *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**

  Help specific to the PDA Module is available from the **PDA Module** book in the **Contents** tab of the *LabVIEW Help*.
- Documentation for the PDA device you use

1 Introduction

The LabVIEW PDA Module is an add-on to LabVIEW that allows you to build applications that run on PDA (Personal Digital Assistant) devices.

The PDA Module extends the capabilities of LabVIEW to allow you to build VIs on a host computer, and then compile and run the VI on a Palm OS or Pocket PC 2002 device. You develop your application using LabVIEW on the host computer. You then build the VI for the PDA and download the resulting application, or PDA VI, to the PDA device to run the application.

PDA VIs

PDA VIs are compiled versions of VIs you can run on a PDA device or emulator. Each PDA VI has a corresponding host VI, which is the VI on the host computer you used to build the PDA VI. PDA VIs have a front panel, but they do not have a block diagram that you can view on the PDA device.

On Palm OS, PDA VIs are `.prc` files. On Pocket PC 2002, PDA VIs are `.exe` files.

PDA Builder

The PDA Builder is a LabVIEW utility you use to compile VIs into PDA VIs that you can run on Palm OS or Pocket PC 2002 devices and emulators. You access the PDA Builder by selecting **Tools»Build for PDA**, which launches the **Build PDA Application** dialog box. Refer to Chapter 2, *Building PDA VIs*, for more information about the PDA Builder.

PDA Emulators

The tools installed with the PDA Module include PDA emulators. Emulators are tools you can use during development to quickly run and test PDA VIs without having to download the PDA VI to the PDA device. The PDA emulator you use depends on the PDA platform you have installed.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help** and clicking the **PDA Module** book in the **Contents** tab, and the documentation included with the emulator for more information about the emulator you use.

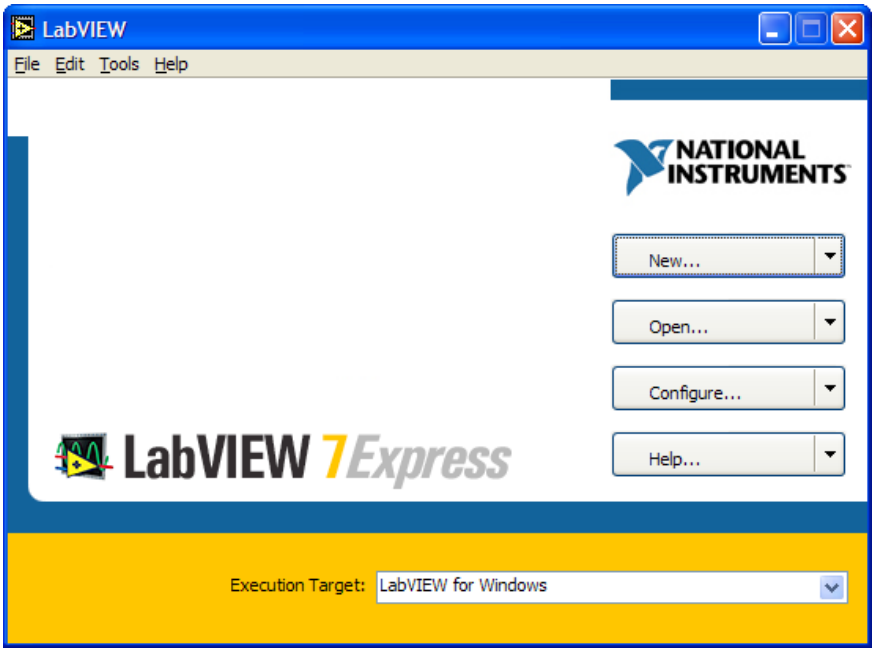
2

Building PDA VIs

This chapter describes how to create applications that run on PDA devices, including compiling the VI to an executable file, or PDA VI, and downloading it to the PDA device.

Selecting the Execution Target

When you launch LabVIEW after installing the LabVIEW PDA Module, LabVIEW contains an extra option for you to define the execution target, as shown in the following figure.



There are three different types of execution targets—LabVIEW for Windows, PDA devices, and PDA emulators. The type of device or emulator is Palm OS or Pocket PC 2002, depending on the PDA Module platform you installed. If you installed the PDA Module for Palm OS and

the PDA Module for Pocket PC 2002, LabVIEW lists separate execution targets for each platform for both PDA devices and emulators.

The execution target you select enables the LabVIEW functionality available for that execution target. For example, because Palm OS devices do not support some front panel and block diagram objects—such as DAQ I/O name controls and ActiveX functions—the **PalmOS Device** execution target includes only the block diagram and front panel objects supported by Palm OS devices.

The execution target also determines where LabVIEW runs the VIs you build. When you select **LabVIEW for Windows** as the execution target and click the **Run** button, the VI runs on the host computer. When you select a PDA device or emulator as the execution target and click the **Run** button, LabVIEW runs the VI in debug mode and connects to the corresponding PDA VI on the selected execution target. Refer to Chapter 3, *Debugging PDA VIs*, for more information about debugging PDA VIs.

To switch execution targets, select an execution target from the **Execution Target** pull-down menu on the **LabVIEW** dialog box.

You also can set the execution target by selecting **Operate»Switch Execution Target**. The options in this dialog box change depending on your choice for the execution target. You can switch execution targets at any time during development.

LabVIEW for Windows

Choose **LabVIEW for Windows** from the **Execution Target** pull-down menu to build VIs for use on the host computer. The **LabVIEW for Windows** execution target uses the default **Controls** and **Functions** palette view.

LabVIEW for Windows is the default execution target.

Palm OS Device

Choose **PalmOS Device** from the **Execution Target** pull-down menu to build VIs for a Palm OS device. When you choose a Palm OS device as the execution target, LabVIEW uses the PDA-4x **Controls** and **Functions** palette view, which contains the front panel and block diagram objects that are available for Palm OS devices and emulators.

When **Palm OS Device** is the execution target, LabVIEW connects to the Palm OS device when you click the **Run** button.

Note You must install `Mathlib.prc` on the Palm OS device to run PDA VIs built for a Palm OS device. Refer to the *LabVIEW PDA Module Release Notes* for information about installing the `Mathlib.prc` file on the Palm OS device.

Palm OS Emulator

Choose **PalmOS Emulator** from the **Execution Target** pull-down menu to build VIs for a Palm OS emulator. When you choose **Palm OS Emulator** as the execution target, LabVIEW uses the PDA-4x **Controls** and **Functions** palette view, which contains the front panel and block diagram objects that are available for Palm OS devices and emulators.

When **Palm OS Emulator** is the execution target, LabVIEW connects to the Palm OS emulator when you click the **Run** button.

Note You must install `Mathlib.prc` on the Palm OS emulator to run PDA VIs built for a Palm OS emulator. Refer to the documentation for your emulator for information about installing the `Mathlib.prc` file on the Palm OS emulator.

Pocket PC 2002 Device

Choose a Pocket PC 2002 device from the **Execution Target** pull-down menu to build VIs for a Pocket PC 2002 device. The **Execution Target** pull-down menu lists each Pocket PC 2002 device that is connected to the host computer, as well as a generic **Pocket PC 2002 (Default Device)** option. When you choose a Pocket PC 2002 device as the execution target, LabVIEW uses the PDA-2x **Controls** and **Functions** palette view, which contains the front panel and block diagram objects that are available for Pocket PC 2002 devices and emulators.

When a Pocket PC 2002 device is the execution target, LabVIEW connects to the Pocket PC 2002 device when you click the **Run** button.

Note PDA VIs built for a Pocket PC 2002 device do not run on Pocket PC 2002 emulators.

Pocket PC 2002 Emulation

Choose **Pocket PC 2002 Emulation** from the **Execution Target** pull-down menu to build VIs for a Pocket PC 2002 emulator. When you choose a Pocket PC 2002 emulator as the execution target, LabVIEW uses the PDA-2x **Controls** and **Functions** palette view, which contains the front panel and block diagram objects that are available for Pocket PC 2002 devices and emulators.

When **Pocket PC 2002 Emulation** is the execution target, LabVIEW connects to the Pocket PC 2002 emulator when you click the **Run** button.

Note PDA VIs built for a Pocket PC 2002 device do not run on Pocket PC 2002 emulators.

Building PDA VIs

You develop PDA VIs on the host computer and build them for the PDA target by selecting **Tools»Build for PDA**, which launches the **Build PDA Application** dialog box.

Note The **Tools»Build for PDA** menu item is disabled when you have **LabVIEW for Windows** selected as the execution target.

The PDA Module compiles the block diagram code into an executable file, or PDA VI, you can run on the PDA target. On Palm OS, PDA VIs are `.prc` files. On Pocket PC 2002, PDA VIs are `.exe` files. You then download the PDA VI to the PDA device by selecting **Operate»Download Application**. You also can use the software provided with the PDA device to download a PDA VI to a PDA device as you do any other application for the PDA device.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help** and selecting the **PDA Module** book in the **Contents** tab, for more information about building PDA VIs.

Transferring Data to and from a PDA Device

You have several options when transferring data between a PDA device and the host computer. Some methods require you to have a connection between the PDA and the computer, and others are wireless.

(Palm OS) File I/O Using the LabVIEW Conduit

The LabVIEW Conduit is a part of the Palm HotSync utility that allows you to copy files that LabVIEW recognizes to and from a Palm OS device.

To customize the behavior of the LabVIEW Conduit, right-click the Palm HotSync icon in the Windows system tray and select **Custom** from the shortcut menu. Select **LabVIEW Conduit** and click the **Change** button. The **LabVIEW Conduit** dialog box appears.

Transferring Files to the Host Computer

To copy files from the Palm OS device to the host computer, select **Handheld overwrites Desktop**. When you sync the Palm OS device, LabVIEW places the files in the location you specify in the path input of the File I/O function you use to create the file in the PDA VI.

Transferring Files to the PDA Device

To copy files to the Palm OS device, click the **Browse** button in the **LabVIEW Conduit** dialog box to select the files. Click the **Add File** button to add a file to the list, select **Desktop overwrites handheld**, and click the **OK** button. LabVIEW copies the files you selected to the Palm OS device the next time you sync it.

Using DatalogMgr

Use the DatalogMgr utility to view a list of files you transferred to the Palm OS device from the host computer or created on the Palm OS device. On the PDA, tap **DatalogMgr** to view a list of all LabVIEW files on the PDA device and the size of each file in number of records or in bytes. DatalogMgr displays where the files will be copied on the host computer when you sync. You also can delete files from DatalogMgr.

Note You must install `DatalogMgr.prc` on the Palm OS device to transfer files to and from the host computer. The LabVIEW Conduit uses the DatalogMgr to detect the presence of LabVIEW files on the PDA. For information about installing the `DatalogMgr.prc` file, refer to the *LabVIEW PDA Module Release Notes*.

(Pocket PC 2002) File I/O Using ActiveSync

On Pocket PC 2002, you use ActiveSync to transfer files between the host computer and the Pocket PC 2002 device. When you connect the Pocket PC 2002 device to the host computer, all files in the `Mobile Device` directory on the host computer are copied to the Pocket PC 2002 device. Any files on the PDA device are automatically copied to their designated directory when you connect the Pocket PC 2002 device to the host computer.

Serial I/O

You can use a serial connection to transfer data between the PDA and the host computer just as you use the serial I/O VIs to transfer data between a computer and an instrument or between two computers.

Use the NI Example Finder, available in LabVIEW by selecting **Help»Find Examples**, to find an example that uses serial I/O to communicate with a host VI.

Infrared

You can use the IrDA functions to transfer data between the PDA and the host computer or another PDA device.

Use the NI Example Finder, available in LabVIEW by selecting **Help»Find Examples**, to find an example that uses IrDA I/O to communicate with a host VI.

Wireless TCP

If you use a PDA device that supports wireless TCP communication, you can use the TCP functions and VIs to transfer data between the PDA device and the host computer or another PDA device. Refer to Chapter 4, *Building Wireless PDA Applications*, for information about creating wireless PDA applications.

Use the NI Example Finder, available in LabVIEW by selecting **Help»Find Examples**, to find examples that use TCP to communicate with a host VI.

3

Debugging PDA VIs

This chapter describes how to debug a PDA VI on a PDA device using its host VI. Debugging PDA VIs is significantly different from debugging VIs on the host computer because PDA VIs do not have a block diagram. You debug PDA VIs while they are connected to the host VIs from which they were built on the host computer.

Enabling Debugging for PDA VIs

By default, the LabVIEW PDA Module does not compile PDA VIs you can debug on the PDA device. To enable debugging, select **Tools»Build for PDA** and click the **Settings** button. In the **Target Settings** dialog box that appears, place a checkmark in the **Generate debug info** checkbox in the **Target Settings** dialog box and build the PDA VI.

Debugging PDA VIs

When you debug a PDA VI, the PDA VI is connected to its host VI using a serial connection on the host computer. You use the block diagram on the host computer as a conduit between the PDA VI running on the PDA device and the host VI.

Note You must use a serial cradle or serial cable to debug a PDA VI on a PDA device. You cannot use a USB connection.

While debugging, the front panel on the PDA VI is fully functional. However, the front panel controls on the host VI have no effect on the PDA VI, nor do the indicators reflect the execution of the PDA VI.

When you debug a PDA VI, you can use all the debugging tools except **Execution Highlighting** on the block diagram of the host VI. You also can insert probes and breakpoints as you do in any other VI.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help** and selecting the **PDA Module** book in the **Contents** tab for more information about debugging PDA VIs.

4

Building Wireless PDA Applications

This chapter describes how you can create services on remote servers and use wireless TCP communication on a PDA device to access the services.

You must have a TCP-capable PDA device to use TCP communication. If you have a PDA device that does not have built-in TCP networking, you might be able to purchase a wireless TCP accessory for the PDA device. Refer to the documentation for the PDA device to determine if you can use TCP networking to build wireless applications.

TCP and UDP Support

The LabVIEW PDA Module supports all LabVIEW TCP functions. Also, you can use the Call By Reference Node to call a VI remotely using a wireless PDA device. To use the Call By Reference Node, you must call the Open Application Reference function and enter `TCP::Ipaddress` in the **machine name** parameter, where *Ipaddress* is the IP address of the remote computer that is running the VIs you want to reference in memory.

Note **(Palm OS)** Palm OS devices do not support the UDP Multicast Open function. Palm OS supports all other UDP functions.

Clients, Services, and Arbitrators

Wireless PDA applications consist of a client, services, and, in some cases, an arbitrator that facilitates communication between the client and the services.

The client is the PDA VI that monitors or controls remote services over the network.

Services are VIs or other applications that perform some task that the client accesses. For example, a service might be a VI that monitors temperature

or tank level or a VI that controls a valve or fan. You build services as you do any other VI. The way you connect to these services depends on the type of wireless application you use.

You use an arbitrator in some wireless applications to facilitate communication between a client and the services on the subnet.

Types of Wireless Applications

There are two main types of wireless PDA applications—direct connection and distributed connection applications.

Direct Connection Applications

Use direct connection applications when you need to connect to the same remote services and you know the location of the remote services.

Direct connection wireless applications consist of a PDA client communicating directly with a service on a server or another device. An example of a direct connection application is a client running on a PDA device that uses the TCP Listen VI to wait for a message from a remote service on the network, and then uses the TCP Read function to read the data that the remote service broadcasts over TCP.

Connecting over TCP or UDP

You can use the TCP or UDP functions to communicate between the client and the service as you do in any other VIs that use TCP or UDP communication. Refer to Chapter 18, *Networking in LabVIEW*, of the *LabVIEW User Manual* for more information about the TCP or UDP communication.

Calling by Reference

You also can call a service by reference using the Call By Reference Node or the PDA Service Request Express VI. VI Server must run on the server to support the Call By Reference Node, and the services available on a server must be in memory so you can reference them.

The PDA Service Request Express VI provides an easy way to connect a PDA client to services by reference. When you place this Express VI on the block diagram, a dialog box appears that prompts you to enter a path to a service. When users run the client on the PDA device, the PDA VI prompts them to enter or retrieve information from a service that the client

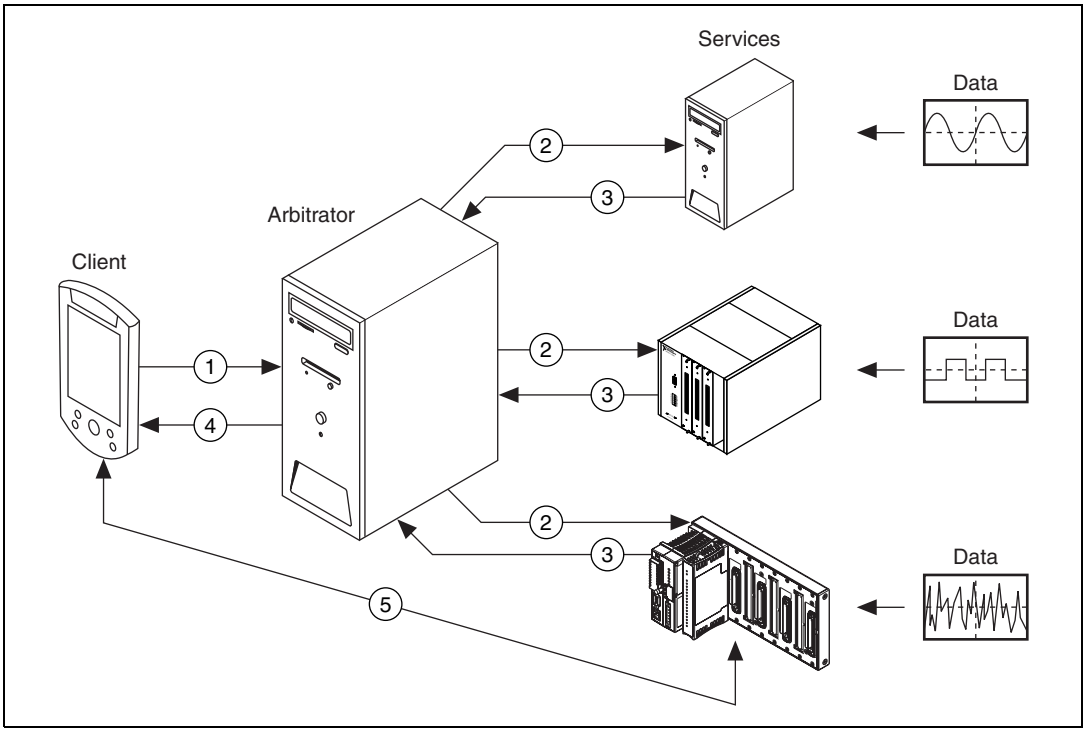
calls by reference on a server. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help** for information about the PDA Service Request Express VI.

Distributed Connection Applications

Use distributed connection applications when you want to have flexible access to several services distributed over a network.

Distributed connection wireless applications consist of a PDA client, remote servers that run service VIs, and an arbitrator VI that facilitates connections between the services and the client. The arbitrator VI can reside on a separate server or on one of the servers where a service resides.

The following figure shows a distributed connection PDA application.



1. The client on the PDA device connects to the server using TCP.
2. The arbitrator sends a UDP broadcast to the local subnet that contains services the PDA client wants to reference. The arbitrator must request services on behalf of the PDA client because some PDA devices do not support UDP broadcasts.

3. When the server that contains a service receives the UDP broadcast, the server checks its available services and replies if it has the services the PDA client requested.
4. The arbitrator replies to the PDA client using TCP with which servers of the local subnet replied as having the services the PDA client requested.
5. The PDA client connects to one of the servers to call a particular service.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about building distributed connection wireless applications. You also can refer to the `DeviceDiscoveryExample.llb` located in the `\examples\PDA` directory for an example of a distributed connection wireless application.

PDA Services Wizard

Use the PDA Services Wizard to build distributed connection applications. The PDA Services Wizard creates an index VI that contains the available services for a server. Select **Tools»PDA Services** to launch the PDA Services Wizard.

You can have more than one index VI on a server. When you create an index VI from the PDA Services Wizard, the only way to edit the index VI is through the wizard.

Caution Do not edit an index VI directly, as doing so can cause the PDA Services Wizard to be unable to load the index VI.

Setting up Servers

In distributed connection applications, you must run the Service Response VI, located in the `vi.lib` directory, on each server that provides services to the client. You must enter the path to the index VI that the PDA Services Wizard created and wire the **machine name** parameter in the Service Response VI before you run it.

On the arbitrator, you must run the Arbitrator Service Query VI, which is located in the `vi.lib\PDA` directory. The Arbitrator Service Query VI facilitates the communication between the client and the services.

When you build the client, you include the PDA Service Query VI as a subVI. The PDA Service Query VI, located in the `vi.lib\PDA` directory, connects to the Arbitrator Service Query VI to initiate the service request.

5

Programming Techniques

This chapter describes limitations you might encounter while creating PDA VIs and ways to work around these limitations. This chapter also describes techniques you can use to create effective PDA VIs.

Refer to the *LabVIEW Help* for additional information about front panel and block diagram features that are unsupported or behave differently on PDA execution tarets than they do on **LabVIEW for Windows**.

Palm OS and Pocket PC 2002 VI Templates

Because PDA devices have smaller screens and a limited set of controls and indicators, the LabVIEW PDA Module includes templates specifically designed for creating PDA VIs. Select **New»PalmScreen** or **PocketPCScreen** from the **LabVIEW** dialog box, or select **File»New** and select **PalmScreen** or **PocketPCScreen** to use these template VIs as the starting point for developing PDA VIs.

(Palm OS) The default fonts used in the PalmScreen template VI are 44-point fonts and the overall size of the front panel is 640 × 640 pixels. When you build a PDA VI for Palm OS devices or emulators, LabVIEW reduces the front panel by a scale factor of four, so fonts appear as 11-point type and the front panel is 160 × 160 pixels.

The PalmScreen template VI also includes a standard Palm OS title bar. Do not put controls or indicators in this region.

(Pocket PC 2002) The default fonts used in the PocketPCScreen template VI are 28-point fonts and the size of the front panel is 480 × 640 pixels. When you build a PDA VI for Pocket PC 2002 devices or emulators, LabVIEW reduces the front panel by a scale factor of two, so fonts appear as 14-point type and the front panel is 240 × 320 pixels.

PDA Device Considerations

The following PDA device considerations can affect the way you design a PDA VI.

- PDA devices have a smaller screen and a significantly slower CPU than desktop computers.
- The main input device is the pen, which makes inputting text slower.
- Many PDA devices do not have floating-point coprocessors, so floating-point operations can be slow because the software performs all floating-point calculations.

PDA Operating System Considerations

Because Palm OS and Pocket PC 2002 are designed for small size and mobility, each environment has fewer features than a desktop computer. As a result of this design, you generally cannot create a VI for a desktop computer and expect it to run on a PDA device without modification.

The following operating system and file system limitations can affect the way you design a PDA VI on both Palm OS and Pocket PC 2002.

- VIs on a desktop computer can go through various states—such as idle, running, and reserved—during which LabVIEW allocates memory for the VI. On PDA devices, a VI is either running or not running. When a PDA VI is not running, the PDA operating system does not allocate memory for the PDA VI, except for uninitialized shift registers.
- PDA devices allocate all the needed memory when a PDA VI runs or is called from another PDA VI. The PDA operating system frees memory when the PDA VI stops or returns. However, the PDA operating system allocates uninitialized shift registers the first time it encounters them and frees them only when the PDA VI stops. Therefore, PDA devices reset controls to their default value each time you run or call a PDA VI, but they do not reset uninitialized shift registers.
- PDA VIs use a just-in-time approach to memory allocation and free memory as soon as the memory is no longer used. As a result, memory usage can depend on the run-time behavior and therefore can be difficult to predict. The advantage to just-in-time memory allocation is that PDA VIs do not allocate unneeded memory.

- PDA VIs ignore all options on the **File»VI Properties** dialog box, with the following exceptions:
  - On the **Execution** page, PDA VIs ignore all priority settings except **subroutine**. PDA VIs set to run in the **subroutine** priority are executed serially.
  - On the **Execution** page, you can place a checkmark in the **Reentrant execution** checkbox to create reentrant PDA VIs.
  - On the **Execution** page, you can place a checkmark in the **Allow debugging** checkbox to enable debugging on PDA devices.
  - On the **Windows Appearance** page, you can click the **Customize** button and place a checkmark in the **Show front panel when called** checkbox in the **Customize Window Appearance** dialog box.

Palm OS Considerations

The following considerations can affect the way you design a PDA VI for Palm OS devices and emulators.

- Palm OS is not multithreaded and has limited memory for programs.
- Palm OS does not use a traditional file system. Instead, it stores data in memory chunks called records that are grouped into databases.
- On Palm OS, all files are implemented as databases and can have a maximum of 64 K records in them, and each record can be a maximum of 64 K in size.
- Each VI that constitutes a PDA VI on Palm OS cannot be larger than 64 K because of the way applications are linked together in Palm OS. However, a top-level PDA VI can contain several subVIs whose total file size is greater than 64 K.

  The size of a VI saved on the hard disk of the host computer is not an accurate indication of how large it will be when you compile it as a PDA VI. When you build a PDA VI, LabVIEW notifies you if the file size is too large. If the VI you want to compile to a PDA VI is too large, break up the VI into smaller subVIs.
- There is a total limit of 64 K on all front panel array data.
- Applications are limited to about 160 K of heap, or dynamic memory space, for everything except code. Some of this space is used by the application to maintain state, so not all of it is available for data. PDA VIs immediately terminate if they run out of memory.

Pocket PC 2002 Considerations

The following considerations can affect the way you design a PDA VI for Pocket PC 2002 devices and emulators.

- PDA VIs are limited to about 32 MB of space for everything except code. Some of this space is used by the application to maintain state, so not all of it is available for data. PDA VIs immediately terminate if they run out of memory.

Front Panel Design

There are several differences between developing a front panel for a host VI and a PDA VI. The following sections describe considerations for window appearance, VI hierarchy, and available controls and indicators.

Window Appearance

Front panels and objects on the front panel behave and appear differently on PDA devices than on VIs on a desktop computer. Consider the following issues when you develop PDA VIs.

- PDA VIs display only one front panel at a time. Calling a subVI that shows its front panel when called causes the subVI front panel to obscure the front panel of the calling VI.
- PDA VIs do not include toolbar icons or standard desktop menus. However, you can use custom run-time menus.
- You cannot tab between controls or use keyboard shortcuts. As you design front panels for PDA VIs, minimize the need for keyboard input.
- When you build the PDA VI on the host computer, LabVIEW scales down all positions, sizes, and fonts by the factor supplied in the **Front panel scale factor** control in the **Target Settings** dialog box.
  - **(Palm OS)** By default, LabVIEW scales front panel objects of a Palm OS PDA VI by four. The standard screen size on a Palm OS device is 160 × 160 pixels, so a 640 × 640 pixel front panel on the host computer works best.

    The standard font on a Palm OS device is 11 points, so use a 44-point font on the front panel of the PDA VIs you build for the Palm OS. The PDA-4x palette view contains controls and indicators with large fonts and sizes you can use to create front panels that work well on Palm OS.

  - **(Pocket PC 2002)** By default, LabVIEW reduces the size of the front panel of a Pocket PC 2002 PDA VI by two. The standard screen size on a Pocket PC 2002 device is 240 × 320 pixels, so a 480 × 640 pixel front panel on the host computer works best.

    The standard font on a Pocket PC 2002 device is 14 points, so use a 28-point font on the front panel of PDA VIs you build for Pocket PC 2002. The PDA-2x palette view contains controls and indicators with large fonts and sizes you can use to create front panels that work well on the Pocket PC 2002.

Hierarchical Limitations

Due to the different application structure on PDA devices, PDA VIs always start from one top-level VI and end when that top-level VI ends. This is different from a typical LabVIEW application where you can start multiple top-level VIs when the application starts and run them simultaneously.

Controls and Indicators

To save memory, most controls and indicators on the PDA-4x and PDA-2x palettes are based on the built-in controls in Palm OS and Pocket PC 2002. The appearance of these built-in controls makes it easy to create a standard user interface, but the controls might not look the same as they do on the host computer.

- The following controls and indicators are built into Palm OS and Pocket PC 2002:
  - Numeric
  - Button
  - Checkbox
  - String and path
  - Enumerated
  - Ring
  - Listbox
  - Table
- Boolean controls have only the **Latch When Released** and **Switch Until Released** mechanical actions. LabVIEW treats other mechanical actions as either **Latch When Released** or **Switch Until Released**.

- If the PDA device supports color, the PDA VI uses the nearest color match. Only some PDA-4x and PDA-2x controls support color. Most built-in controls use the default system colors and you cannot change them.
- You can place bitmaps on front panels of PDA VIs. You also can use the 1-bit, 4-bit, and 8-bit pixmap functions in the picture control. However, LabVIEW does not scale bitmap images when you build the PDA VI. The bitmaps retain their original size to avoid the aliasing effects that can result if they are reduced in size.

Unsupported Front Panel Objects

If you place unsupported controls on a front panel, LabVIEW discards them when you build the PDA VI and they are not shown on the PDA device. Also, PDA VIs ignore unsupported features of supported controls. The following front panel objects are not available on PDA devices:

- Tab controls
- Multicolumn listboxes
- Intensity graphs and charts
- Digital waveform graphs
- 3D graphs
- Polar plots
- Smith plots
- Minmax plots
- Radar plots
- Picture rings
- Text and picture rings
- I/O controls and indicators
- ActiveX controls and indicators
- Variant controls and indicators

Block Diagram Programming

This section describes techniques you can use for developing block diagrams for PDA VIs.

- Because PDA VIs are compiled applications for the PDA device, you cannot view the block diagram on the PDA device. You can view and edit the block diagram from the host VI from which you built the PDA VI.
- Most basic LabVIEW functions are supported. However, the underlying behavior, particularly with respect to memory usage, is completely different. Refer to the *Application Design and Performance Considerations* section for information about memory usage.
- You can use the host computer to remotely debug a PDA device placed in its cradle. Refer to Chapter 3, *Debugging PDA VIs*, for information about debugging techniques.

Unsupported Block Diagram Objects

The following block diagram objects are unavailable on PDA devices.

- Waveform data types and all related VIs and functions
- **(Palm OS)** Data acquisition and instrument I/O features
- **(Pocket PC 2002 Emulator)** TCP Open Connection and UDP Open functions
- Probes with pictures or paths

Application Design and Performance Considerations

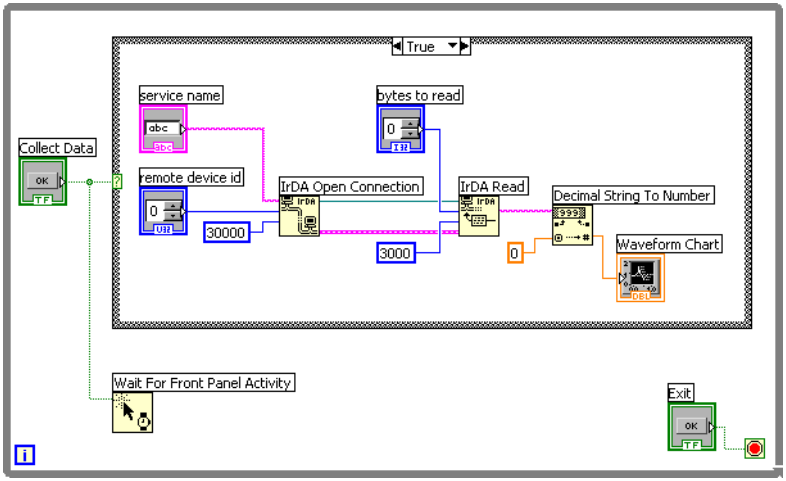
This section describes some techniques for designing PDA VIs based on the restrictions and limitations stated earlier in this chapter.

Good Application Design

All PDA devices run on battery power, which is a limited resource. Users typically run PDA applications for short intervals and then expect the device to switch to low power mode until the next time they need it. Most PDA devices are designed to run for eight hours, but that time is usually in small increments spread over a period of a month.

To conserve battery resources, include a Wait For Front Panel Activity function or an Event structure in a PDA VI. In the following example, the

Wait For Front Panel Activity function creates a wait state to save battery power. Adding a wait state allows the PDA to go to sleep if the user does not tap the screen for some duration specified in the preferences.



Performance Considerations

LabVIEW optimizes PDA VIs for data size, speed, and code size in that order.

Design PDA VIs to have wide, shallow hierarchies of VIs. Because PDAs allocate memory only when a PDA VI is running, nested VI calls use large amounts of memory and might cause the PDA VI to fail. Wide, shallow hierarchies have only a few VIs running at any one time and use less memory.

Even though Palm OS does not support multiple threads, it does use cooperative multitasking. Therefore, you can design PDA VIs to have parallel loops on the block diagrams. One loop runs for a time slice (approximately 50 ms), then another loop runs for a time slice. PDA VIs automatically use cooperative multitasking when you use two or more nodes that are not connected by a wire. However, PDA VIs with parallel loops run slower than ones that are serialized.

If you set the execution priority of a PDA VI to **subroutine**, all execution is serialized, and the PDA VI runs much faster. However, if you set the execution priority of a PDA VI to **subroutine**, the user interface does not

get any CPU time, and the application might become unresponsive. You can address this by placing a Wait Until Next ms Multiple function on the block diagram.

A

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at ni.com for technical support and professional services:

- **Support**—Online technical support resources include the following:
  - **Self-Help Resources**—For immediate answers and solutions, visit our extensive library of technical support resources available in English, Japanese, and Spanish at ni.com/support. These resources are available for most products at no cost to registered users and include software drivers and updates, a KnowledgeBase, product manuals, step-by-step troubleshooting wizards, conformity documentation, example code, tutorials and application notes, instrument drivers, discussion forums, a measurement glossary, and so on.
  - **Assisted Support Options**—Contact NI engineers and other measurement and automation professionals by visiting ni.com/support. Our online system helps you define your question and connects you to the experts by phone, discussion forum, or email.
- **Training**—Visit ni.com/custed for self-paced tutorials, videos, and interactive CDs. You also can register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, NI Alliance Program members can help. To learn more, call your local NI office or visit ni.com/alliance.

If you searched ni.com and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.